HỌC MÁY

TS. NGUYỄN HUYỀN CHÂU
AI LAB - TLU

L01: Giới thiệu về Học máy

Giới thiệu về học máy

Giới thiệu bài toán phân lớp

Quy trình phát triển học máy

Mô hình học

Outline

Giới thiệu về học máy

Giới thiệu bài toán phân lớp

Quy trình phát triển học máy

Mô hình học

HỌC MÁY LÀ GÌ?

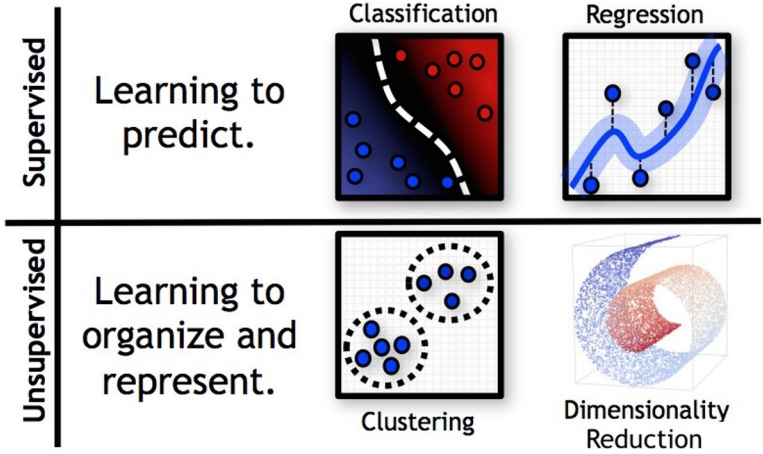
Các quan niệm về học máy

Samuel (1959): "Học máy là một lĩnh vực nghiên cứu cho phép máy tính học mà không cần lập trình tường minh."

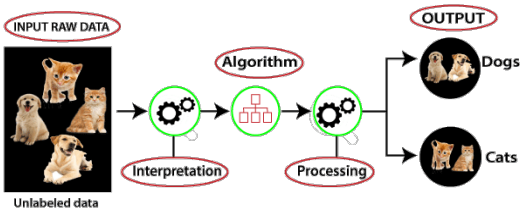
Mitchell (1997): "Một chương trình máy tính được cho là đã học được kinh nghiệm E xét trên nhiệm vụ T cùng độ đo P, nếu hiệu quả máy thực hiện nhiệm vụ T đo bởi P được cải thiện nhờ E."

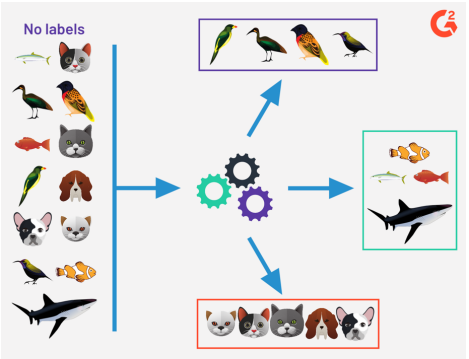
Trong học máy, "kinh nghiệm E" chính là nằm trong "tập dữ liệu D".

Các dạng nhiệm vụ học máy



Ví dụ các dạng nhiệm vụ học máy





Học có giám sát

Học không giám sát

Ví dụ các nhiệm vụ học máy

- **Phân lớp (Classification):** Phân mỗi mẫu dữ liệu vào một lớp cụ thể (Lớp: một giá trị lấy từ một tập hữu hạn các giá trị).
- **Hồi quy (Regression):** Gắn mỗi mẫu dữ liệu với một giá trị số thực (Dự đoán giá nhà, giá cổ phiếu, ..)
- **Xếp hạng (Ranking):** Xếp thứ tự mẫu dữ liệu theo tiêu chí cho trước (Tìm các trang web liên quan và xếp thứ tự tìm kiếm)
- **Phân cụm (Clustering):** Phân mẫu dữ liệu thành các cụm gồm các mẫu giống nhau (Khác gì nhiệm vụ phân lớp?)
- **Giản ước chiều (Dimensionality reduction):** Giảm số lượng thuộc tính mô tả dữ liệu nhằm giảm độ phức tạp lưu trữ và tính toán.

Outline

Giới thiệu về học máy

Giới thiệu bài toán phân lớp

Quy trình phát triển học máy

Mô hình học

Nhiệm vụ phân lớp

Định nghĩa:

Cho vector đặc trưng $\boldsymbol{x} \in \mathrm{X} = \mathbb{R}^D$ mô tả một đối tượng biết chắc thuộc một lớp trong số C lớp của tập các lớp Y, đoán xem đối tượng đó thuộc lớp nào.

Ví dụ: Từ thời tiết, sức gió, nhiệt độ dự đoán trận đấu có diễn ra hay không.

- Thời tiết, sức gió, nhiệt độ : Các yếu tố làm nên vector đặc trưng $\boldsymbol{x}$
- {Diễn ra, Không diễn ra}: 2 lớp đầu ra, làm thành tập đầu ra Y với C = 2

Bộ phân lớp (classifier)

Định nghĩa:

Cho tập dữ liệu D = $\{(\boldsymbol{x^i}, \boldsymbol{y^i}), i = 1{:}m\}$ với $\boldsymbol{x^i} \in \mathrm{X} = \mathbb{R}^D$ là một vector đặc trưng và $\boldsymbol{y^i} \in \mathrm{Y}$ là một nhãn, thì một bộ phân lớp là một hàm $h: \mathbb{R}^D \to Y$ mà dự đoán một nhãn $h(\boldsymbol{x}) = \hat{\boldsymbol{y}}$ cho mỗi vector đặc trưng $\boldsymbol{x}$ bất kỳ.

Lưu ý: Hàm h đưa được dự đoán cho cả $\boldsymbol{x} \notin \mathrm{D}$, nghĩa là với cả dữ liệu chưa thấy

i	Thời tiết	Sức gió	Lớp
1	Mưa	Mạnh	Không
2	Nắng	Yếu	Có
3	Bình thường	Mạnh	Không
4	Mưa	Yếu	Không
5	Bình thường	Mạnh	Không
6	Mưa	Mạnh	Có

Ví dụ: Cho hai bộ phân lớp sau:

H1 (weather, wind) = (weather = *sunny*)

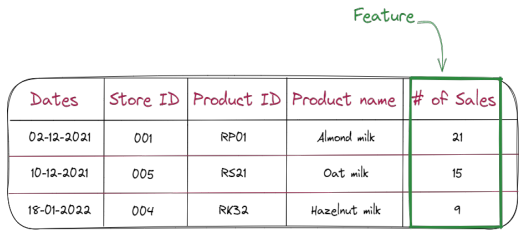
H2 (weather, wind) = (wind = *weak*) OR
(weather = *normal)*

Hỏi: Nếu trời mưa, gió yếu thì trận đấu diễn ra hay không theo H1? Theo H2?

Mô hình hoá bài toán phân lớp

no_samples: Số lượng mẫu (n). Mỗi mẫu được dung cho mỗi quy trình(ví dụ: phân loại). Mẫu có thể là tài liệu, hình ảnh, tệp âm thanh, video, hàng trong cơ sở dữ liệu hoặc dòng một tệp CSV.

no_features: Số lượng các đặc trưng (d) hoặc thuộc tính. Các đặc trưng thường là một cột chứa các kiểu giá trị định lượng hoặc định tính



Mô hình hoá bài toán phân lớp

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ x_{31} & x_{32} & \cdots & x_{3d} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$$

$$\mathbf{y}^T = [y_1, y_2, y_3, \cdots y_n]$$

Các thuật ngữ

- **Ví dụ/quan sát/điểm dữ liệu**/đối tượng/trường hợp/hiện thể/bản ghi: một đơn vị đầu vào.
- **Các đặc trưng** (features)/**thuộc tính** (attributes): các trường thông tin của một ví dụ, các trường này thường hợp thành một vector
- **Nhãn**: là mục tiêu cần dự đoán (VD lớp trong bài toán phân lớp hay giá trị trong bài toán hồi quy). Từ các cặp <đầu vào, nhãn mẫu>, học cách dự đoán nhãn cho đầu vào mới.

Outline

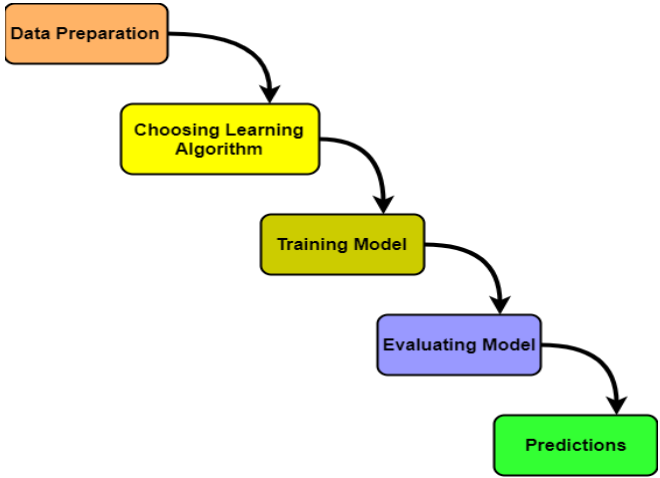
Giới thiệu về học máy

Giới thiệu bài toán phân lớp

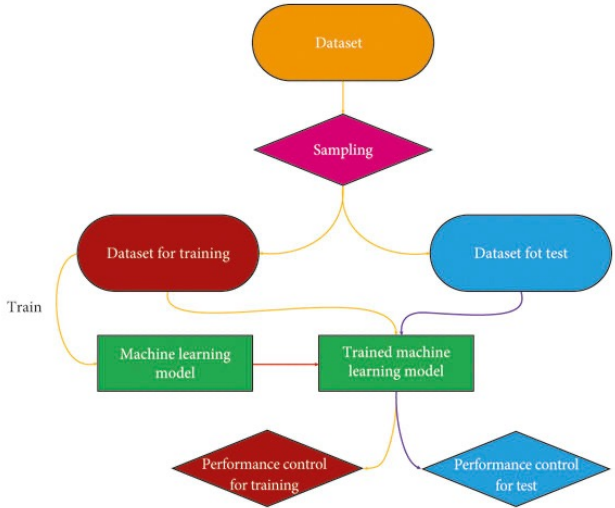
Quy trình phát triển học máy

Mô hình học

Quy trình phát triển học máy



Xây dựng bài toán trong thư viện Scikit-learn

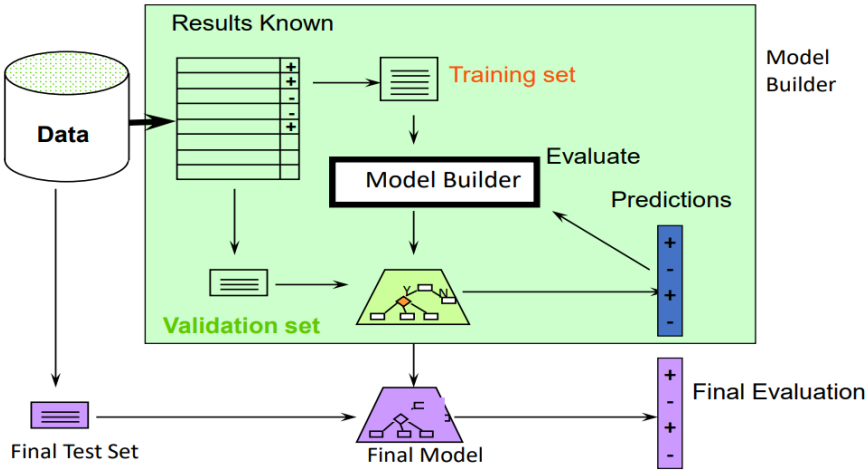


Phân chia tập dữ liệu

Phân chia tập dữ liệu huấn luyện

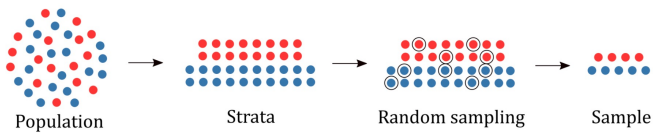
- Dữ liệu: thường được phân chia thành 3 tập con, dùng với 3 mục đích khác nhạu
    - **Training data** - Dữ liệu huấn luyện (có gắn nhãn, để học luật cơ bản).
    - **Validation data** - Dữ liệu xác thực (có gắn nhãn, để học luật có tính tổng quát)
    - **Testing data** - Dữ liệu kiểm tra (có gắn nhãn song giấu không cho thấy, để ước đoán chất lượng khi triển khai thực tế)

Tập huấn luyện, xác thực, kiểm tra



Xác thực chéo (Cross-validation)

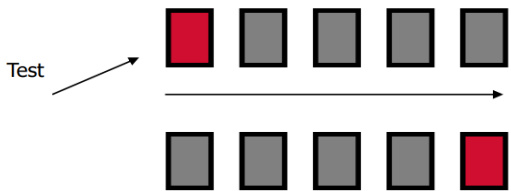
- Cross-validation giúp các tập test khỏi chồng lấn
  - Bước đầu tiên: Chia dữ liệu thành *k* tập con cùng cỡ
  - Bước thứ hai: mỗi tập con lần lượt được dùng để validate và phần còn lại để train
- Đây gọi là k-fold cross-validation
- Sai số ước lượng được lấy trung bình để cho ra sai số ước lượng chung
- Các tập con thường được phân tầng (stratified) trước khi thực thi cross-validation. VD: hình dưới mô tả sự phân tầng theo các nhãn, rồi chia tập theo từng tầng



Cross-validation

- Chia dữ liệu thành các tập cùng cỡ

- Dành một tập để test và phần còn lại để xây model



- Lặp lại

Thêm về cross-validation

- Phương pháp chuẩn cho đánh giá: phân tầng ten-fold cross-validation
- Tại sao lại 10? Nhiều thí nghiệm công phu đã chỉ ra đây là lựa chọn tốt nhất để ra một ước lượng chính xác
- Phân tầng làm giảm phương sai của ước lượng

Xây dựng mô hình
Ví dụ: K Nearest Neighbors

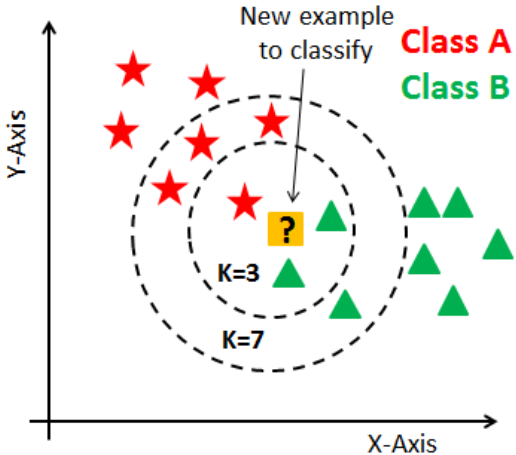
K Nearest Neighbors (K láng giềng gần nhất)

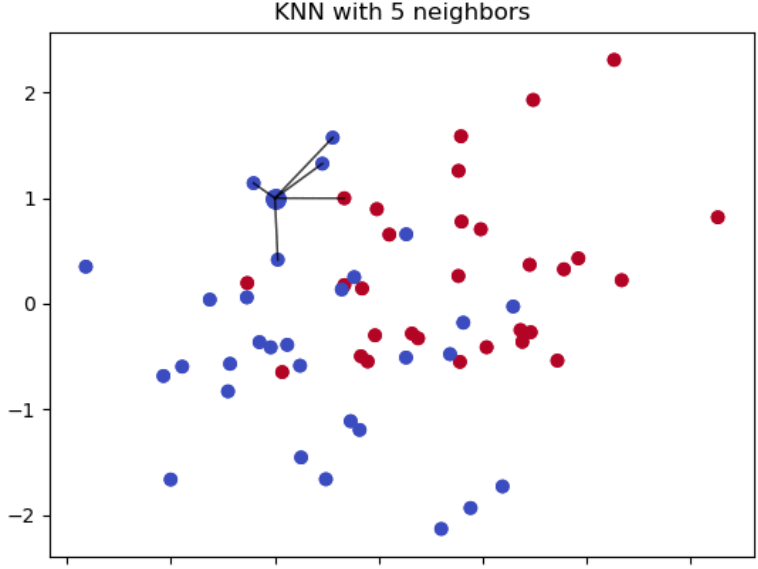
KNN là một bộ phân loại phi tham số và sẽ lưu trữ cả tập huấn luyện D. Trong KNN, đầu ra cho mỗi điểm dữ liệu mới $\boldsymbol{x}$ được bình bầu đa số từ K điểm gần $\boldsymbol{x}$ nhất, là một tập $N_K(\boldsymbol{x})$ xét theo hàm khoảng cách d nào đó $d : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$

$$f_{KNN}(\boldsymbol{x}) = \underset{c}{\operatorname{argmax}} \sum_{i \in N_K(\boldsymbol{x})} [\mathrm{y}_i = c]$$

Để cài đặt KNN cần biết hàm khoảng cách d và số điểm láng giềng gần nhất K

Minh hoạ KNN




KNN with 5 neighbors
2
1
0
−1
−2
−3
−2
−1
0
1
2
3

Brute Force KNN

- Cho một hàm khoảng cách *d* bất kỳ, ta có thể chạy KNN kiểu brute force bằng cách tính mọi khoảng cách $d_i = d(\boldsymbol{x^i}, \boldsymbol{x}^*)$ từ một điểm đích $\boldsymbol{x}^*$ tới tất cả các điểm huấn luyện $\boldsymbol{x^i}$.
- Sau đó sắp xếp các khoảng cách này rồi chọn *K* điểm có khoảng cách với đích nhỏ nhất để tạo thành tập láng giềng $N_K(\boldsymbol{x}^*)$.
- Thực hiện hoàn toàn tương tự với hàm tương đồng, chỉ là chọn *K* điểm tương đồng nhất.
- Khi đã chọn xong *K* láng giềng, phần còn lại khá đơn giản

Các độ đo khoảng cách

Định nghĩa: Khoảng cách Minkowski (các chuẩn ℓ_p)

Cho hai vector dữ liệu $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$, khoảng cách Minkowski Distance với tham số p (chuẩn ℓ_p) là một metric định nghĩa như sau:

$$d_p(\boldsymbol{x}, \boldsymbol{x}') = ||\boldsymbol{x} - \boldsymbol{x}'||_p = \left(\sum_{i=1}^{D} |x_d - x'_d|^p\right)^{\frac{1}{p}}$$

Các trường hợp riêng: khoảng cách Euclide ($p = 2$), khoảng cách Manhattan ($p = 1$ và khoảng cách Chebyshev ($p = \infty$).

Ví dụ: KNN

i	Nhiệt độ	Độ ẩm	Lớp
1	30	10	Có
2	24	15	Không
3	31	17	Có
4	29	19	Không
5	22	12	Có

Dữ liệu về chơi tennis: Với dữ dữ liệu mới x = {27,11}, k = 3. Dự đoán xem x thuộc lớp nào, sử dụng khoảng cách Manhattan.

- d[1] = |data[i]-x| = |30-27| + |11-10| = 3+1= 4
- d[2] = |data[i]-x| = |24-27| + |15-10| = 3+5=8
- d[3] = |data[i]-x| = |31-27| + |17-10| = 4+7=11
- d[4] = |data[i]-x| = |29-27| + |19-10| = 2+9=11
- d[5] = |data[i]-x| = |22-27| + |12-10| = 5+2=7

Với k =3, dữ liệu x gần với điểm 1,2,5 = {Có, Không, Có} -> x thuộc lớp Có.

Bài tập: KNN

i	x	y	Kết quả bầu cử
1	1	1	Joe Biden
2	1	2	Donald Trump
3	2	6	Joe Biden
4	3	4	Donald Trump
5	8	1	Donald Trump
6	2	4	Joe Biden

Dữ liệu toạ độ về kết quả bầu cử tổng thống

Ta có dữ liệu mới: a = {x = 10,y=5}, k = 5. Vậy người a sẽ bầu cho vị tổng thống nào?

- Dùng khoảng cách Manhattan

Huấn luyện mô hình

Huấn luyện mô hình

- Thực hiện fit dữ liệu vào mô hình sau khi xây dựng để đảm bảo mô hình chạy đúng, có thể học ra kết quả
- Sau đó, dần cải thiện kết quả bằng tìm siêu tham số phù hợp. Để tối ưu thời gian và mô hình đạt kết quả tốt, ta có thể thực hiện sử dụng hàm GridsearchCV để tìm ra tham số phù hợp

Đánh giá

Tỷ lệ sai phân lớp

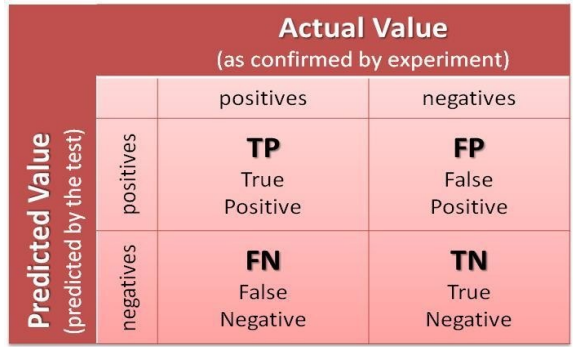
- Độ đo hiệu năng cơ bản cho bài toán phân lớp: tỷ lệ sai
  - *Success*: đối tượng được phân đúng lớp
  - *Error*: đối tượng bị phân sai lớp
  - *Error rate*: tỷ lệ các đối tượng *Error* trên tổng số đối tượng

$$Error\ Rate = \frac{Error}{total\ number\ of\ test\ íntances}$$

- *error rate của tập train:* quá lạc quan
  - Luôn tìm được khuôn mẫu ngay cả trong dữ liệu ngẫu nhiên

Confusion matrix - Ma trận nhầm lẫn

- **True Positive (TP):** Dự đoán là positive và đúng (còn gọi là hit)
- **True Negative (TN):** Dự đoán là negative và đúng (còn gọi là từ chối đúng)
- **False Positive (FP):** Sai lầm loại I; dự đoán là positive nhưng sai (cảnh báo sai)
- **False Negative (FN):** Sai lầm loại II; dự đoán là negative nhưng sai (còn gọi là miss)



Các độ đo khác

- **Sensitivity** hay **True Positive Rate** (TPR) (còn gọi là **hit rate**, **recall**): $TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$
- **Specificity** (SPC) hay **True Negative Rate** (TNR): $TNR = SPC = \frac{TN}{N} = \frac{TN}{FP + TN}$
- **Precision** hay **Positive Predictive Value** (PPV): $PPV = \frac{TP}{TP + FP}$
- **Negative Predictive Value** (NPV): $NPV = \frac{TN}{TN + FN}$

Các độ đo khác

- **Accuracy (ACC)**

$$ACC = \frac{TP + TN}{P + N}$$

- **F_1 Score** (harmonic mean of precision recall)

$$F_1 = \frac{2TP}{2TP + FP + FN} = \frac{2}{\frac{1}{TPR} + \frac{1}{PPV}}$$

Ví dụ

STT	Kết quả dự đoán	Kết quả thực tế
1	xanh	xanh
2	xanh	xanh
3	đỏ	xanh
4	đỏ	đỏ
5	xanh	đỏ
6	đỏ	xanh
7	xanh	đỏ

Yêu cầu tính các thông tin sau cho cả nhãn + và nhãn -:

- TP, TN, FP, FN,
- accuracy, độ nhạy, độ đặc hiệu
- F1 score

Lưu ý:

- Tính theo từng lớp: Coi lớp đó là dương tính, các lớp còn lại đều là âm tính, từ đó tính TP, TN, FP, FN, độ nhạy, độ đặc hiệu, F1 score
- Sai số và độ chính xác thì không cần tính theo lớp mà tính trên tổng.

Outline

Giới thiệu về học máy

Giới thiệu bài toán phân lớp

Quy trình phát triển học máy

Mô hình học

Mô hình hóa việc học

- Không gian giả thuyết
- Mô hình bài toán
- Thuật toán học

Mô hình hoá việc học

- **Không gian:** Không gian đầu vào X, không gian kết quả Y. Tập huấn luyện sẽ gồm các ví dụ tạo từ cặp < đầu vào $\boldsymbol{x}$, kết quả $\boldsymbol{y}$ > với $\boldsymbol{x} \in \mathrm{X}$ và $\boldsymbol{y} \in \mathrm{Y}$
- **Giả thuyết**: Mỗi giả thuyết $h : \mathrm{X} \to \mathrm{Y}$ sẽ gán một đầu vào $\boldsymbol{x} \in \mathrm{X}$ với một kết quả dự đoán $h(\boldsymbol{x}) = \hat{\boldsymbol{y}} \in \mathrm{Y}$.
- **Họ giả thuyết:** Tập H gồm các ánh xạ h từ X đến Y mà bộ học lựa chọn khảo sát
- **Đầu ra dự đoán: Hàm mất mát/hàm sai số:** Một ánh xạ $\mathcal{L}: \mathrm{Y} \times \mathrm{Y} \to \mathbb{R}$.
  - $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$: sai số giữa giá trị dự đoán $\hat{\boldsymbol{y}}$ so với giá trị chân lý $\boldsymbol{y}$.
  - Với bài toán phân loại nhị phân, $\mathcal{L}$ là sai số giữa 0 và 1, nên $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \mathbb{I}_{\hat{\boldsymbol{y}} \neq \boldsymbol{y}}$
  - Với bài toán hồi quy và có $\mathrm{Y} \subseteq \mathbb{R}$ thì $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \hat{\boldsymbol{y}} - \boldsymbol{y}$

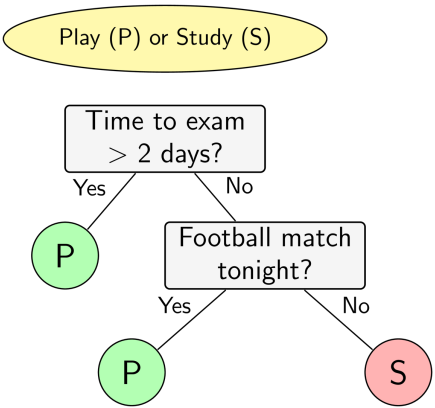
Thiết lập mô hình học có giám sát

Tập huấn luyện: Là tập mẫu cỡ m được lấy theo kiểu i.i.d. từ tập $\mathrm{X} \times \mathrm{Y}$:

$$\mathrm{D} = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\}$$

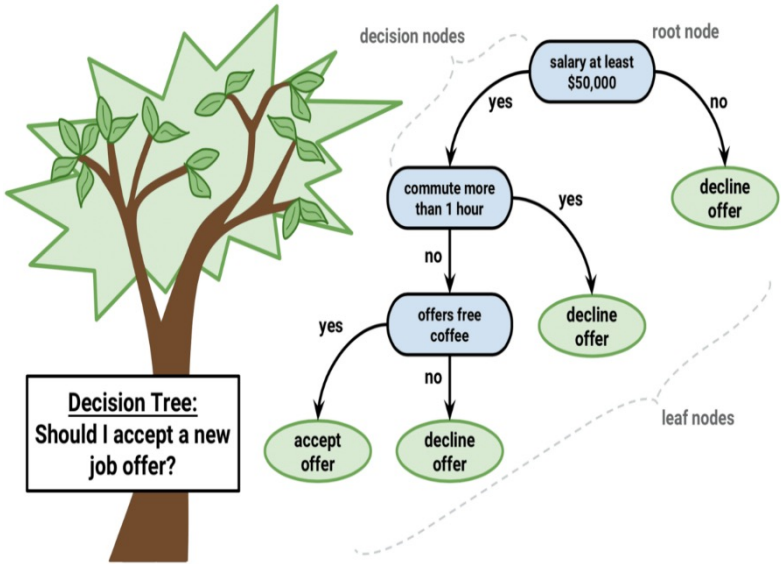
Bài toán: Tìm ra giả thuyết $h \in \mathrm{H}$ mà có tổng sai số toàn tập mẫu nhỏ.

Thuật toán học: Cây quyết định

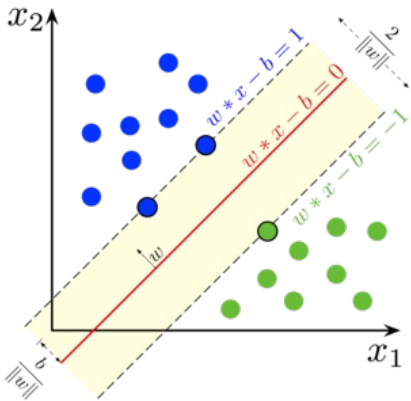


- Một cây quyết định cổ điển sẽ phân loại các ví dụ nhờ vào những kiểm tra logic được tổ chức thành một cấu trúc cây nhị phân
- Mỗi nút nội chứa một luật kiểm tra có dạng ($x_d < t$) hay ($x_d = t$), nhằm so sánh một chiều dữ liệu đơn lẻ d với một giá trị t và phân một ví dụ về cây con trái hay phải tuỳ theo kết quả.
- Mỗi ví dụ sẽ duyệt dọc cây từ gốc tới một lá nào đó. Mỗi nút lá tương ứng một phân lớp mà ví dụ sẽ được xếp vào.

Thuật toán học: cây quyết định

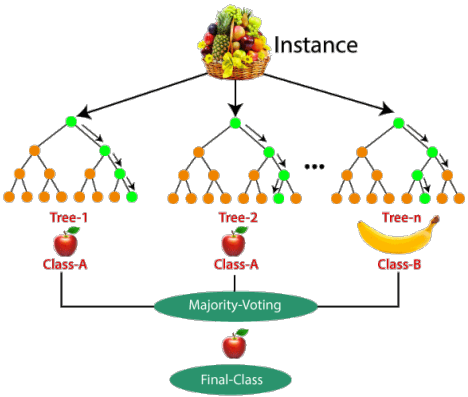


Thuật toán học: Support vector machine



- Thuật toán tìm ra hyper plane để phân chia các điểm dữ liệu
- Margin là khoảng cách biên từ siêu phẳng đến các điểm dữ liệu

Thuật toán học: Random Forest



- Rừng được xây dựng từ nhiều cây quyết định có cùng bộ siêu tham số
- Kết quả cuối của rừng được thực hiện bằng tổng bình bầu đa số từ nhiều cây